A Channel Access Protocol for Multihop Wireless Networks with Multiple Channels

Andrew Muir **J.J. Garcia-Luna-Aceves**
Computer Engineering Department
University of California
Santa Cruz, CA 95064

Abstract

The Group Allocation Multihop Multiple Access (GAMMA) protocol is presented; this protocol schedules data traffic over a multihop, multi-channel, wireless network. GAMMA provides excellent performance and remains stable under all network load levels by dividing the channel into cycles; each cycle is composed of a combination of contention and data slots. Every station in the network has a unique channel for receiving data. Each station maintains a set of stations called the "transmission group", only members of this group are allowed to transmit data collision-free to the station maintaining the group.

1 Introduction

Existing media access control (MAC) protocols can be divided into two groups: contention-based and contention-free. A contention-based protocol such as CSMA [10, 14] requires a station to compete for control of the transmission channel each time it sends a message; this strategy is very efficient when the network load is low. However, as the traffic level rises, it becomes increasingly difficult to transmit data as stations prevent each other from taking control of the channel; this causes the average delay to grow rapidly. Consequently, contention-based schemes cannot provide stability under heavy network loads. Many contention-free protocols have been proposed; these protocols, which use polling, token passing or reservations, our stable under heavy loads, but are not efficient under light loads.

We describe and analyze a MAC protocol that we call "Group Allocation Multihop Multiple Access" (GAMMA) which is designed for use in an ad-hoc wireless network operating in unlicensed bands and using commercial radios. In such networks, no base station can be used to implement polling, and because stations move in and out of the network, it is difficult to implement token passing. Consequently, most ad-hoc networks use a contention-based protocol for controlling channel access, and therefore, are not stable under all conditions.

The most difficult problem associated with ad-hoc networks is that of "hidden terminals" [18]; hidden terminals exist when two or more stations can communicate with a third station, but not with each other. Several approaches have been used to solve this problem. The Busy Tone Multiple Access (BTMA) protocol [18] divides the transmission channel into a busy-tone channel and a message channel. BTMA uses a base-station to transmit a busy-tone signal for as long at it senses carrier on the message channel. However, this requires that the base station is line of site with every station in the network, which limits the utility of this protocol.

The Multiple Access Collision Avoidance (MACA) protocol [9] attempts to solve the hidden terminal problem by using a Request To Send (RTS) / Clear To Send (CTS) message exchange. MACA does not use carrier sensing because carrier sensing does not provide information on the state of the channel at the destination node. Recently, floor acquisition multiple access (FAMA) protocols [5] have been proposed to allow for the collision-free transmission of packet trains. FAMA protocols require a successful two-way handshake between sender and receiver before the sender transmits any data, the sender transmits a RTS packet and waits for the receiver to respond with a CTS packet before sending its data packet.The handshake ensures that all stations agree to listen for the sender's data packets, i.e., it allows the sender to acquire control of the channel (called the floor). This is a form of dynamic channel reservations, but it cannot provide stability under high loads because stations must contend for the floor each time they transmit a data packet or packet train. Other protocols such as [2, 3, 4, 7, 17] use some form of an RTS/CTS exchange to control the channel.

Several reservation based protocols have been proposed such as [1, 13, 8, 11, 12] which provide stability at high load levels, and efficient service at low load levels. Resource auction protocols, i.e. [1, 12] require a significant amount of overhead for each auction period and are difficult to implement. On the other hand, PRMA [8] is relatively easy to implement but uses a fixed frame length which can lead to starvation if the number of active stations is large, and PRMA uses a data packet to make a channel reservation which is inefficient in case of an unsuccessful reservation. These protocols all require a base station, and do not operate in a network with hidden terminals. In contrast, GAMMA does not use a base station and uses multiple channels to ensure collision-free data transmissions in a network with hidden terminals. Also, GAMMA does not have a single point of failure, i.e. a base station, and it is able to function correctly in any ad-hoc network.

In section 2 we describe GAMMA in detail; section 3 presents the throughput and delay of our simulation of GAMMA, and section 4 offers some concluding remarks.

2 Protocol Description

GAMMA is designed to operate in an ad-hoc wireless network with multiple transmission channels. It is not necessary for the network to be fully connected; consequently, "hidden terminals" may exist. Each station in the network listens to a unique channel; any transmissions to a station must be sent on this channel, and access to the channel is controlled by the station. GAMMA does not place any restrictions on the physical layer, i.e., a transmission channel may be: a dedicated frequency, a unique code, or a distinct hopping sequence.

Every station maintains a set of stations called the "transmission group"; only members of this group are allowed to transmit data to the station. Once a station is a group member, it is able to transmit data collision-free until it no longer has data to send. A station with a message to send must join the destination's transmission group before it transmits data. Likewise, before a station receives data, the source must first join the station's transmission-group.

2.1 Cycle Structure

Each transmission channel is divided into a series of cycles, and a cycle is composed of a variable number of slots. There are two types of slots, contention and data. During a contention slot, a station with a message to send uses an RTS/CTS message exchange to gain membership in the destination's transmission group. When a station is admitted into this group, it is assigned a data slot which it can use for as long as it has data to send. It should be noted that a station may either send or receive during a data slot; as we will explain later in this section, a data slot may also be unallocated. An example transmission channel is shown in fig. 1.

Because the number of data slots per cycle depends upon the number of stations with data to transmit to the station maintaining the cycle, the size of the cycle varies with the volume of network traffic directed at the station. Therefore, it is possible for two stations to have different cycle lengths. In order for these stations to complete an RTS/CTS exchange their contention slots must be aligned; therefore, the number of data slots between contention slots is fixed at some number q. If the number of data slots in a cycle is less than q, then the interval between contention slots includes data slots from multiple cycles. Consequently, the number of contention slots per cycle varies.

If a station transmits data during a data slot, it sends information on its cycle in the header of each data packet; this information includes the number of data slots in the cycle, and which data slots are unallocated. If a station receives data during a data slot, it sends this same information in its acknowledgement of the data packet.

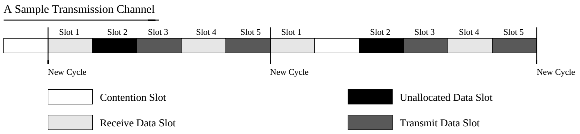

Figure 1: GAMMA: In this example there are five data slots in a cycle; however, the interval between consecutive contention slots is six. Because of this difference, the number of contention slots per cycle varies; the first cycle does not have a contention slot while the second cycle has one. A data slot may be used to transmit or receive data or it may be unallocated.

2.2 Group Admission

After a message to be transmitted arrives at a station, the station must join the destination's transmission group; the station transmits an RTS (on the destination's channel) during the next contention slot. If the destination receives the RTS, it responds with a CTS, which signifies that the source of the RTS has been admitted into the transmission group. Both the RTS and the CTS contain information on their source's cycle, e.g. the number of data slots, and which data slots are unallocated; this information is used to determine which data slot should be used to transmit/receive data. The destination may not receive the RTS for one of two reasons, either another station transmits an RTS during the same slot (causing a collision of RTSs), or the destination transmits an RTS on another stations channel, i.e. it is not listening to its own channel. In either case, the station transmitting the unsuccessful RTS backs-off, and attempts to retransmit at some random point in the future.

Both the source and the destination compare the cycle information in the RTS/CTS to information on their own cycle, in order to determine which slot to use for transmitting/receiving data. Once a slot is assigned to a connection, it is critical that the source and destination maintain synchronization around this slot; therefore, both stations must have the same cycle length for the duration of the connection. The cycles do not have to be aligned, i.e. start and end at the same time; however, if their lengths are not identical, the cycles will shift, and the slots allocated from both the source and destination's cycles will no longer be aligned. Consequently, the stations will not able to exchange data.

If the source and destination's cycle length are not the same, then the station with the shorter cycle must add additional data slots. After the successful RTS/CTS exchange, the stations process each of their data slots once, and then add the new data slots (if necessary). As a part of processing each data slot, the station informs its member stations that it is adding data slots; this information is contained in the header of either the data packet or the return acknowledgement. A station which recognizes that another station is increasing its cycle length, must increase its own cycle length in order to remain aligned. Consequently, a ripple effect is produced, in which an expanding circle of stations increase their cycle length.

There are three rules that a station uses to determine which slot to assign to a connection.

1. If the cycle lengths of both the source and destination are the same, i.e. they have the same number of data slots, and they both have an unallocated data slot which is aligned with the other, then these unallocated slots are chosen for the connection. An example of this is shown in fig. 2.
2. The source and destination have different cycle lengths, but at least one of the unallocated or additional slots from the shorter cycle is aligned with an unallocated slot from the longer cycle. In which case the unallocated slot from the larger cycle is chosen. Fig. 3 shows an application of this rule.
3. There are no unallocated slots in the longer cycle which are aligned with an unallocated or additional (if one of the cycles had a shorter length) slot in the shorter cycle,

as in fig. 4. In this case, the station with the longer cycle adds an additional slot. This additional slot (from the longer cycle) is guaranteed to be aligned with one of the new slots in the shorter cycle, and so is chosen as the slot for use by the connection. Note that the shorter cycle must be increased by an additional slot in order to maintain synchronization.

The three rules are applied in order with the slot being chosen by the first rule which finds a match. The slot is determined by both stations independently. Because both stations have current information on the other station's cycle, each will decide upon the same slot.

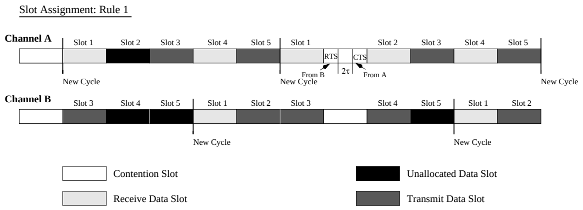

Figure 2: GAMMA: This example shows an application of rule 1. Station B sends an RTS on station A's channel, and station A responds with a CTS on the same channel. Since both stations have the same cycle length, and unallocated slot 2 from station A and 4 from station B are aligned, no additional slots are required. These slots are chosen for the connection, as can be seen in the second cycle in which slot 2 has been assigned to receive data and slot 4 has been assigned to transmit data. These assignments are maintained for the duration of the connection.

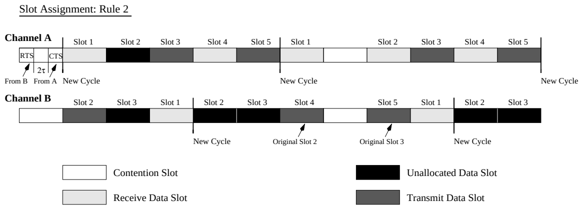

Figure 3: GAMMA: In this figure, rule 2 is used to choose a pair of slots for a new connection. Station A has 5 slots per cycle, and station B has 3; therefore, station B must add two unallocated cycles in order to maintain synchronization with station A. Before these slots are inserted, each remote station must be informed that the cycle length has increased. Therefore, the slots are inserted after one pass through the cycle; this pass does not necessarily start at slot 1 (in this example it starts at slot 2).

2.3 Changing the Cycle Length

When a station is notified by one of its neighbors that the cycle length of the neighbor is increasing, the station must increase its cycle by the same amount, in order to keep its slot aligned with the slot of the neighbor. The station informs each of the neighbors it is connected to that it is also increasing its cycle length, and then it adds new (unallocated) data slots.

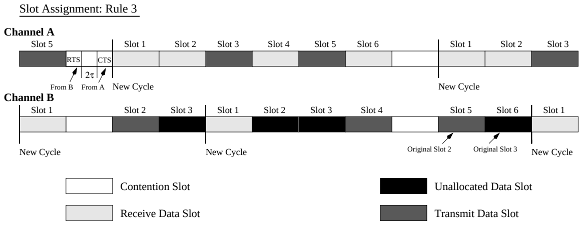

Figure 4: GAMMA: Rule 3 is used to determine the slot assignment for this example. As in the previous figure, station A has 5 slots per cycle, and station B has 3. Also, station A does not have any unallocated slots, which means that station A must add an additional slot in order to receive data from station B. Consequently, Station B must add 3 slots in order to be synchronized with station A. One of these additional slots is guaranteed to be aligned with station A's additional slot; in this case, slot 4 is aligned, and is allocated for transmitting data to station A.

GAMMA includes an additional flag in the header of each data/acknowledgement packet which informs a station when it is able to reduce its cycle length. This flag is set in each transmission if either a station does not have an unallocated data slot, or if the most recent data/acknowledgement packet it received from any of its neighbors has this flag set. When a station exchanges data/acknowledgement packets with another station, it will leave this flag clear if the other station is its only neighbor without an unallocated slot. For this flag to be clear, all of the inter-connected stations in the network must have at least one unallocated data slot. If this flag remains clear for some predetermined number of cycles, a station will attempt to remove one data slot; following an unallocated data slot, the station informs each of its neighbors via data/acknowledgement packets that the unallocated data slot has been removed.

After a station receives notification that one of its neighbors has removed a data slot, i.e. that it has gone from m to $m-1$ data slots, the station must also remove an unallocated data slot, and inform its own neighbors of the change. However, it is not possible to remove a slot until each neighbor has been informed because their schedule is based on a cycle of m slots. In order to maintain synchronization with the neighbor that initially shrunk its cycle, the station temporarily (for one cycle) removes that neighbor's data slot, which ensures that every neighbor is aware of the change in cycle length, and that the station has only $m-1$ slots; thus, only the station which initiated the removal of a data slot misses a data slot, and all stations maintain their synchronization. In the following cycle an unallocated slot is removed, and the neighbor is able to use its slot again. Because a station misses data slots when it initiates a cycle contraction, a station does not react too quickly to the opportunity to remove a data slot. Instead, it waits for a specific number of cycles, which prevents the station from responding to short-term fluctuations in the cycle length. It is possible that a station may assign its last unallocated slot while another station is attempting to shrink its own cycle; GAMMA handles this situation in the same way it does an expanding cycle.

2.4 Group Departure

A station maintains its membership in a transmission group for as long as it has data to transmit. When a station is ready to leave a group, it informs the destination of its intent by setting a flag in the header of its data packet. Both stations mark the slot as being unallocated; this slot is then available for assignment to a new group member. If a data slot is idle for some specified number of cycles, the destination assumes that the source has failed, and the slot is added to the pool of unallocated slots.

3 Performance Results

We show the throughput (for a single transmission channel) and average delay of a simulation of GAMMA over a variety of network configurations. The capacity of the channel is 1 Mb/s, and the arrival rate corresponds to the number of messages to be transmitted that arrive at each station within single frame. The throughput is measured as the amount of data transmitted over a single channel. Because the target radios are half-duplex, the highest possible throughput would be 0.5.

Fig. 5 plots the throughput and delay of GAMMA as a function of arrival rate; the size of a data packet is 200 bytes, and the average number of packets per message is 100. Each curve represents a different value for the maximum number of neighbors. For a given arrival rate, when a station has more neighbors, the arrival rate of RTSs to that station will also be greater. Consequently, at low arrival rates, the throughput is higher for networks with more neighbors per station because it is more likely that a station will have at least one member in its transmission group. However, because the length of each cycle also increases, the average delay grows with the number of neighbors.

In Fig. 6, we compare the throughput and delay of GAMMA for a variety of message sizes. The maximum number of neighbors is 50, the packet size is 200 bytes, and the average number of packets per message ranges from 20 to 1000. The graphs show that the throughput increases and the delay decreases as the average message size increases. When the number of packets per message is high, the average group size is increased because each successful RTS reserves a position in the group for more cycles; as in fig. 5 the throughput increases as the group size increases.

4 Conclusion

We have presented a new channel access protocol for multi-channel, multi-hop wireless networks called the Group Allocation Multihop Multiple Access (GAMMA) protocol. GAMMA is stable and efficient under all load levels.

A unique channel is allocated to each station in the network; each channel is divided into variable length cycles, and a cycle is composed of data and contention slots. A data slot can be used to either transmit or receive data, and is assigned to a specific station. Each node maintains a "transmission group" which consists of all stations able to transmit data to the node. A station with a message to transmit must first join the destination's transmission group by successfully completing an RTS/CTS exchange with the destination during one of the contention slots. Once a station has been successfully added to the transmission group, it can send data collision-free during each cycle, until the message is complete. When a network is lightly loaded, GAMMA behaves much like CSMA. As the load in the channel grows, the cycle size increases up to a maximum, after which GAMMA becomes in effect TDMA, giving every station that is part of the transmission group a "slot" in each cycle.

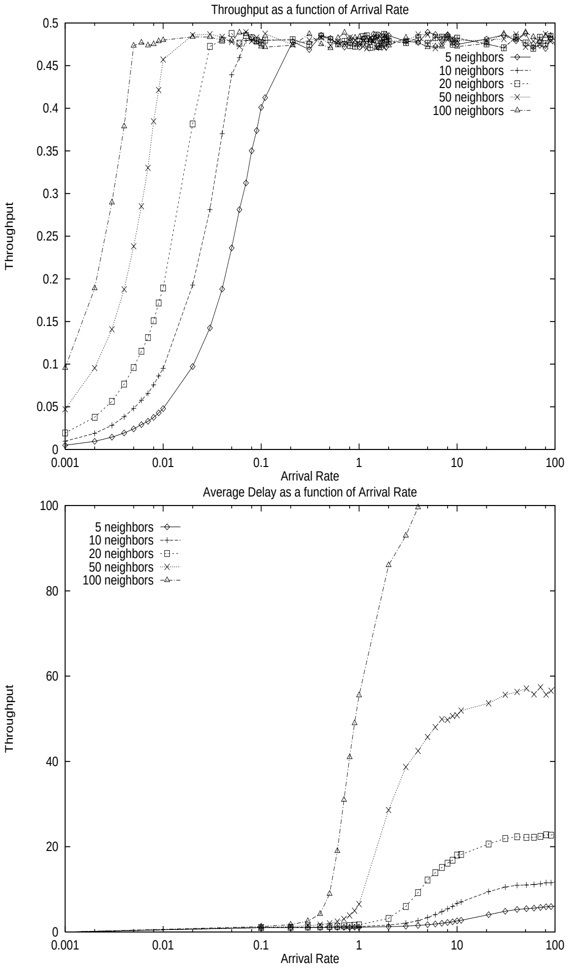

Figure 5: A comparison of the throughput and average delay of GAMMA for a variety of maximum number of neighbors.

References

[1] N. Amitay and S. Nada *"Resource Auction Multiple Access (RAMA) for Statistical Multiplexing of Speech in Wireless PCS" Proc. ICC '93, May 1993*

[2] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's", *Proc. ACM SIGCOMM '94*

[3] K. Chen, "Medium Access Control of Wireless LANs for Mobile Computing", *IEEE Network*, vol. 8, no. 5

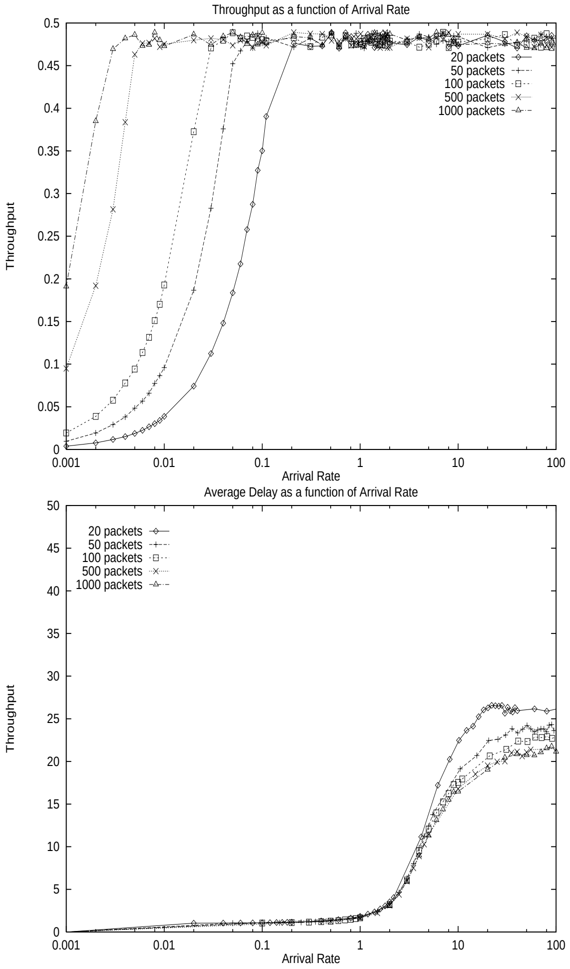

Figure 6: A comparison of the throughput and average delay of GAMMA for a variety of average message sizes.

[4] A. Colvin, "CSMA with Collision Avoidance", *Computer Commun.*, vol. 6, no. 5

[5] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks", *Proc. ACM SIGCOMM 95.*

[6] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "FAMA-PJ: A Channel Access Protocol for Wireless LANs", *Proc. First Int'l Conf. on Mobile Computing and Networking 1996*

[7] *P802.11-Unapproved Draft: Wireless LAN Medium Access Control (MAC) and Physical Specifications*, IEEE, January 1996

[8] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard and B. Ramamurthy *"Packet Reservation Multiple Access for Local Wireless Communications" IEEE Transactions on Communications, August, 1989*

[9] P. Karn, "MACA- a New Channel Access Method for Packet Radio", *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*

[10] L. Kleinrock and F. Tobagi, "Packet Switching in Radio Channels: Part I - Carrier Sense Multiple Access Modes and their Throughput-Delay Characteristics", *IEEE Trans. Commun.*, vol. COM-23, no. 12

[11] Dong Geun Jeong, Chong-Ho Choi, Wha Sook Jeon "Design and Performance Evaluation of a New Medium Access Control Protocol for Local Wireless Data Communications" IEEE/ACM Transactions on Networking, December 1995

[12] M. Karol and Chin-Lin I *"A Protocol for Fast Resource Assignment in Wireless PCS" IEEE Transactions on Vehicular Technology, August 1994*

[13] M. Karol, Z. Liu and K.Y. Eng *"An Efficient Demand-Assignment Multiple Access Protocol for Wireless Packet (ATM) Networks" Wireless Networks, October 1995*

[14] B. Leiner, D. Nielson, and F. Tobagi, eds., *Proc. of the IEEE* vol. 75

[15] W. Lo and H. Mouftah, "Carrier Sense Multiple Access with Collision Detection for Radio Channels", *IEEE Int'l Commun. and Energy Conf.*

[16] A. Muir and J.J. Garcia-Luna-Aceves, "Group Allocation Multiple Access with Collision Detection", *Proc. IEEE INFOCOM '97*

[17] R. Rom, "Collision Detection in Radio Channels", *Local Area and Multiple Access Networks*, Computer Science Press, 1986

[18] F. Tobagi and L. Kleinrock "Packet Switching in Radio Channels: Part II - the Hidden Terminal Problem in Carrier Sense Multiple-Access Modes and the Busy-Tone Solution", *IEEE Trans. Commun.* vol. COM-23, no. 12